CSC Frameworks Assignment: Turn-Based Game with Crow + Qt

Overview

In this assignment, you will design and implement a small turn-based adventure game using a **Crow (C++) backend server** and a **Qt (C++) graphical frontend**. Your player can move, encounter enemies, attack them, use potions, and level up. This is a foundational exercise in building real-time connected applications using modern C++ frameworks.

Goals

- Practice building and integrating a RESTful backend using Crow
- Learn to create dynamic UIs with Qt and communicate using JSON
- Model a persistent game state shared across requests
- Demonstrate ability to test and explain full-stack application behavior

1 Requirements

You must implement the following features:

1. Movement System

- Add directional buttons (North, South, East, West) to the Qt frontend
- Each button sends a POST request to `/move` with a direction
- The backend updates and returns the player's position

2. Enemy Encounter

- At a fixed position (e.g., `x=2, y=1`), an enemy appears
- The server sets `enemy_alive = true` and sends an encounter message
- The frontend displays "An enemy appears!" to the user

3. Turn-Based Combat

- Add an "Attack" button in Qt
- Each attack reduces enemy HP; if still alive, the enemy strikes back
- When enemy HP reaches 0, the player levels up and enemy resets

4. Potion System

- A "Use Potion" button increases player HP up to a maximum
- The server handles this and returns the updated HP

5. State Display

- The Qt UI should show messages, player HP, level, and position
- Make use of QLabel and layout tools to clearly organize feedback

2 Deliverables

You must submit the following items to the provided GitHub Classroom repository:

1. `crow_main.cpp` — backend server source file
2. `qt_main.cpp` — frontend UI and logic source file
3. `README.md` — instructions for building and running both components
4. **Panopto video** — 5–7 minute screen recording that:
   - Shows the application running
   - Walks through key parts of your code
   - Explains how the client and server interact

3 Grading Rubric (100 points)

1. UI and Backend Design Clarity Clear structure, logical UI layout, clean game state modeling	25 pts
2. Application Implementation Functioning features (movement, combat, potion, level-up)	25 pts
3. Code Quality and Readability Consistent style, helpful comments, good separation of concerns	25 pts
4. Panopto Video Explanation Explains logic, demonstrates interaction, is clear and complete	25 pts

Bonus (up to 6 extra points)

- **Visited Tile Tracking (2%)** – Tiles the player has moved through appear visually different in the UI.
- **Minimap View (2%)** – A smaller view of the full scene is displayed beside the main game view.
- **Animated Movement (2%)** – Player movement is animated using Qt transitions instead of jumping.